

WHAT IS CLAIMED IS:

1 1. A method for use by a first process executing in a computer system for
2 interacting with a second process executing in the computer system, the method comprising:
3 during a startup sequence of the second process, creating a copy of a global
4 notification hook of the first process in the second process;
5 using the copy of the global notification hook, detecting an occurrence of a
6 triggering message passed between an operating system and a thread of the second process;
7 in response to detecting the occurrence of the triggering message, determining
8 whether subsequent messages passed between the operating system and the thread of the
9 second process should be monitored; and
10 in the event that the subsequent messages should be monitored, activating a
11 thread-level message hook within the thread of the second process, wherein the thread-level
12 message hook is configured to monitor the subsequent messages.

1 2. The method of claim 1, wherein the thread-level message hook is
2 further configured to cause an action to occur in response to a specified subsequent message.

1 3. The method of claim 2, wherein the action includes creating a visual
2 effect for a window of the second process.

1 4. The method of claim 1, wherein the thread-level message hook is
2 configured so as not to affect operation of a third process executing concurrently with the
3 second process in the computer system.

1 5. The method of claim 1, wherein the triggering message is a window
2 creation message.

1 6. The method of claim 5, wherein the act of determining whether
2 subsequent messages should be monitored includes determining whether the window creation
3 message relates to a window of interest.

1 7. The method of claim 6, wherein the window creation message relates
2 to a window of interest unless one or more of the following conditions obtains: (a) the
3 window creation message does not relate to a visible window; (b) the created window has a

4 window type designated by a user as not being of interest; and (c) the created window has a
5 window type that is incompatible with the thread-level message hook.

1 8. The method of claim 1, wherein the first process is a desktop
2 management process.

1 9. The method of claim 8, wherein the second process is an application
2 process.

1 10. The method of claim 1, further comprising, during a startup sequence
2 of the first process:
3 detecting a third process executing in the computer system;
4 inserting a copy of the global notification hook into the third process; and
5 broadcasting a private startup message to the copy of the global notification
6 hook in the third process.

1 11. The method of claim 10, wherein, in response to the private startup
2 message, the copy of the global notification hook executes acts of:
3 determining whether subsequent messages passed between the operating
4 system and a thread of the third process should be monitored; and
5 in the event that the subsequent messages should be monitored, activating a
6 thread-level message hook within the thread of the third process.

1 12. The method of claim 11, wherein determining whether subsequent
2 messages passed between the operating system and the thread of the third process should be
3 monitored includes:
4 identifying a previously created window of the third process; and
5 determining whether the previously created window is of interest, wherein
6 subsequent messages should be monitored in the event that the previously created window is
7 of interest.

1 13. The method of claim 1, wherein the act of activating the thread-level
2 message hook includes mapping executable code for the thread-level message hook into an
3 address space of the second process.

1 14. The method of claim 1, wherein the act of creating the copy of the
2 global notification hook includes mapping executable code for the global notification hook
3 into an address space of the second process.

1 15. The method of claim 1, wherein the act of detecting the occurrence of
2 the triggering message includes receiving message data of the triggering message.

1 16. The method of claim 15, wherein the message data of the triggering
2 message is provided to the copy of the global notification hook concurrently with a
3 transmission of the triggering message to the thread of the second process.

1 17. The method of claim 16, wherein the second process has a
2 process-specific message queue that receives the transmitted message data of the triggering
3 message.

1 18. A method for use by a first process executing in a computer system for
2 interacting with a second process executing in the computer system, the method comprising:
3 during a startup sequence of the first process, creating a copy of a global
4 notification hook of the first process in the second process; and
5 broadcasting a private startup message from the first process to the copy of the
6 global notification hook;
7 wherein, in response to the private startup message, the copy of the global
8 notification hook executes acts of:
9 determining whether subsequent messages passed between the
10 operating system and a thread of the second process should be monitored; and
11 in the event that subsequent messages should be monitored, activating
12 a thread-level message hook within the thread of the second process, wherein the thread-level
13 message hook is configured to monitor the subsequent messages.

1 19. The method of claim 18, wherein the thread-level message hook is
2 configured so as not to affect operation of a third process executing concurrently with the
3 second process in the computer system.

1 20. The method of claim 18, wherein the thread-level message hook is
2 further configured to cause an action to occur in response to a specified subsequent message.

1 21. The method of claim 18, wherein the action includes creating a visual
2 effect for a window of the second process.

1 22. The method of claim 18, wherein the global notification hook
2 determines that subsequent messages should be monitored in the event that a window of
3 interest exists in the second process.

1 23. The method of claim 22, wherein a window existing in the second
2 process is of interest unless one or more of the following conditions obtains: (a) the window
3 is not a visible window; (b) the window has a window type designated by a user as not being
4 of interest; and (c) the window has a window type that is incompatible with the thread-level
5 message hook.

1 24. The method of claim 18, wherein the first process is a desktop
2 management process.

1 25. The method of claim 24, wherein the second process is an application
2 process.

1 26. The method of claim 18, wherein the act of activating the thread-level
2 message hook includes mapping executable code for the thread-level message hook into an
3 address space of the second process.

1 27. A computer program product comprising:
2 a computer readable medium encoded with program code for a global
3 notification hook of a first process, wherein the program code for the global notification hook
4 is adapted to be copied into a second process during a startup sequence of the second process,
5 the program code for the global notification hook including:

6 program code for detecting an occurrence of a triggering message in
7 the second process;

8 program code for determining, in response to detecting the occurrence
9 of the triggering message, whether subsequent messages passed between the operating
10 system and a thread of the second process should be monitored; and

11 program code for activating a thread-level message hook within the
12 thread of the second process in the event that the subsequent messages should be

13 monitored, wherein the thread-level message hook is configured to monitor the
14 subsequent messages.

1 28. The computer program product of claim 27, wherein the computer
2 readable medium is further encoded with program code for the thread-level message hook,
3 the program code for the thread-level message hook including:
4 program code for controlling an action to be taken in the event that a selected
5 subsequent message is detected.

1 29. The computer program product of claim 27, wherein the computer
2 readable medium comprises a magnetic storage medium encoded with the program code.

1 30. The computer program product of claim 27, wherein the computer
2 readable medium comprises an optical storage medium encoded with the program code.

1 31. The computer program product of claim 27, wherein the computer
2 readable medium comprises a carrier signal encoded with the program code and adapted for
3 transmission via a network.

1 32. A computer program product comprising:
2 a computer readable medium encoded with program code for a global
3 notification hook of a first process, wherein the program code for the global notification hook
4 is adapted to be copied into a second process during a startup sequence of the first process,
5 the program code for the global notification hook including:
6 program code for broadcasting a private startup message from the first
7 process to the copy of the global notification hook;
8 program code for determining, in response to the private startup
9 message, whether subsequent messages passed between the operating system and a
10 thread of the second process should be monitored; and
11 program code for activating a thread-level message hook within the
12 thread of the second process, in the event that the subsequent messages should be
13 monitored, wherein the thread-level message hook is configured to monitor the
14 subsequent messages.

1 33. The computer program product of claim 32, wherein the computer
2 readable medium is further encoded with program code for the thread-level message hook,
3 the program code for the thread-level message hook including:
4 program code for controlling an action to be taken in the event that a specified
5 subsequent message is detected.

1 34. The computer program product of claim 32, wherein the computer
2 readable medium comprises a magnetic storage medium encoded with the program code.

1 35. The computer program product of claim 32, wherein the computer
2 readable medium comprises an optical storage medium encoded with the program code.

1 36. The computer program product of claim 32, wherein the computer
2 readable medium comprises a carrier signal encoded with the program code and adapted for
3 transmission via a network.